This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Canceled)

2. (Previously Presented) The method of claim 31 wherein the first computer entity encrypts the code ID of the attestation message according to a key available to the second computer entity, the method further comprising the second computer entity decrypting such encrypted matter.

3. (Previously Presented) The method of claim 31 wherein the second computer entity consumes the attestation message by application of same to a verifying function that automatically verifies the attestation message based on a format thereof and that extracts relevant information from such verified attestation message for use by the second computer entity.

4. (Previously Presented) The method of claim 31 wherein the first computer entity is a part of a computing device, and the second computer entity decides based on the code ID in the attestation message whether the first computer entity can be trusted, and also decides based on a certificate chain of the message whether the computing device can be trusted, the certificate chain leading back to a trusted root authority.

5-6. (Canceled)

7. (Previously Presented) The method of claim 4 wherein the second computer entity determines that the code ID is a known code ID and that the first computer entity can be trusted based on such code ID.

8. (Previously Presented) The method of claim 4 wherein the second computer entity determines from the certificate chain whether the computing device of the first computer entity

should be trusted to instantiate and operate the first computer entity in a trusted manner and should be trusted to calculate the code ID properly.

9. (Previously Presented) The method of claim 8 wherein the second computer entity determines that each certificate in the certificate chain is not on a do-not-trust list.

10. (Previously Presented) The method of claim 31 wherein the trust message includes a symmetric key (K) that the first and second computer entities employ to encrypt and decrypt messages therebetween.

11. (Previously Presented) The method of claim 10 wherein the symmetric key (K) is encrypted according to a public key (PU-1) to result in (PU-1(K)), the second computer entity obtaining (PU-1) from the certificate chain of the attestation message, and wherein the first computer entity obtains the symmetric key (K) from the received trust message by applying a private key (PR-1) corresponding to (PU-1) to (PU-1(K)) to result in (K).

12. (Previously Presented) The method of claim 31 wherein the trust message further includes an identification of a cryptographic algorithm to be employed in connection with the first secret.

13. (Canceled)

14. (Previously Presented) The method of claim 31 wherein the trust message further includes relevant trust data encrypted according to a key available to the first computer entity, and wherein the first computer entity decrypts the encrypted trust data by applying the key thereto.

15. (Canceled)

16. (Previously Presented) The method of claim 31 wherein the second computer entity creates the trust message by application of a sealing function that automatically produces the trust message in an appropriate format that is accessible to the first computer entity.

17. (Canceled)

18. (Previously Presented) The method of claim 31 wherein prior to the first computer entity transmitting the attestation message, the first computer entity sends a can-attest message to the second computer entity, the can-attest message stating that the first computer entity can send an attestation message but that the first computer entity would like to know from the second computer entity whether such an attestation message is required by such second computer entity and if so any requirements that such second computer entity has with regard to such attestation message, the method further comprising the second computer entity sending an attestation-wanted message to the first computer entity in response to the can-attest message, the attestation-wanted message stating that the second computer entity does in fact require an attestation message from the first computer entity and that the attestation message as sent by the first computer entity must adhere to certain requirements as defined in such attestation-wanted message, whereby the first computer entity thereafter sends the attestation message in accordance with the requirements stated in the attestation-wanted message.

19. (Previously Presented) The method of claim 30 further comprising:

the first computer entity constructing, in accordance with the requirements stated in the attestation-wanted message, the attestation message to be delivered to the server, the attestation message including a code identifier (code ID) representative of the first computer entity and data relevant to the purpose of the trust-based relationship;

the first computer entity appending a digital signature to the attestation message and a certificate chain leading back to a trusted root authority, the signature being based on the code ID and data thereof and being verifiable based on a security key included in the certificate chain, the

certificate chain including at least one certificate therein proffering trustworthiness of the first computer entity;

the first computer entity sending the attestation message to the server and the server receiving same, whereby the server entity verifies the signature of the received attestation message based on the included security key, whereby alteration of the code ID or data of the attestation message should cause the signature to fail to verify, the server based on such a failure dishonoring such attestation message, the server decides whether to in fact enter into the trust-based relationship with the first computer entity based on the code ID and the data in the attestation message, the server upon deciding to in fact enter into the trust-based relationship with the first computer entity constructs a trust message to be delivered to the first computer entity, the trust message establishing the trust-based relationship and including therein a secret to be shared between the first computer entity and the server, where such shared secret allows such first computer entity and the server to communicate in a secure manner, and the server sends the trust message to the first entity and the first entity receiving same; and

the first computer entity obtaining the shared secret in the trust message and employing the shared secret to exchange information with the server according to the established trust-based relationship with such server.

20. (Previously Presented) The method of claim 19 wherein the code identifier (code ID) is calculated from a digest of the first computer entity, whereby alteration of the first computer entity causes the code ID to change.

21. (Previously Presented) The method of claim 20 wherein the code identifier (code ID) is calculated from the digest of the first computer entity and from security information relating thereto, whereby alteration of the first computer entity or the security information causes the code ID to change.

22. (Canceled)

23. (Previously Presented) The method of claim 19 further comprising a code ID calculator of the first computer entity that is used for calculating the code ID, the code ID calculator operating in a trusted manner in a computing device.

24. (Canceled)

25. (Previously Presented) The method of claim 19 wherein the first computer entity creates the attestation message by application of the code ID and data thereof to a quoting function that automatically produces the attestation message in an appropriate format that is accessible to the server.

26. (Previously Presented) The method of claim 19 wherein the server constructs a trust message including therein a shared secret comprising a symmetric key (K) that the first computer entity and the server employ to encrypt and decrypt messages therebetween, the symmetric key (K) being encrypted according to a public key (PU-1) to result in (PU-1(K)), the server obtaining (PU-1) from the certificate chain of the attestation message, the method comprising the first computer entity obtaining the symmetric key (K) from the received trust message by applying a private key (PR-1) corresponding to (PU-1) to (PU-1(K)) to result in (K).

27. (Previously Presented) The method of claim 19 wherein the server constructs a trust message further including relevant trust data encrypted according to a key available to the first computer entity, the method comprising the first computer entity decrypting the encrypted trust data by applying the key thereto.

28. (Previously Presented) The method of claim 19 wherein the first computer entity consumes the trust message by application of same to an unsealing function that automatically extracts the shared secret and other relevant information from such trust attestation message for use by the first computer entity.

29. (Currently Amended) The method of claim 19 whereby the trust message is a first trust message and the shared secret is a first shared secret, and whereby the server constructs a second trust message to be delivered to the first computer entity, the second trust message including therein a second secret to be shared between the first computer entity and the server, where such second shared secret allows such first computer entity and the server [[s]] to communicate in a secure manner, and the server sends the second trust message to the first computer entity and the first computer entity receives same, the method further comprising the first computer entity obtaining the second shared secret in the trust message and employing the second shared secret to exchange information with the server, whereby the first shared secret is no longer valid.

30. (Previously Presented) A method of establishing trust between a first computer entity and a server, the method comprising:

the first computer entity seeking a granting of trust from the server by sending an inquiry in the form of a can-attest message to the server, the can-attest message stating that the first computer entity can send an attestation message but that the first computer entity would like to know from the server whether such an attestation message is required, and if so any requirements that such server has with regard to such attestation message; and

the server sending an attestation-wanted message to the first computer entity in response to the can-attest message, the attestation-wanted message stating that the server does in fact require an attestation message from the first computer entity and that the attestation message as sent by the first computer entity must adhere to certain requirements as defined in such attestation-wanted message,

one of the certain requirements being that the attestation message is to include a code identifier (code ID) associated with the first computer entity and calculated by using a security ID associated with the first computer entity, the security ID including security information relating to the first computer entity, the security information being expressed as a number of name-value security attribute parameters, the first computer entity being an executable and referring to the parameters in the security information in the security ID to determine whether

particular security behavior is allowed, the code identifier (code ID) being representative of the first computer entity and calculated as a one-way hash of a combination of the executable of the first computer entity and the security ID so that modification of the security information in the security ID causes the calculated code ID to change and the server can interpret the change as an indication that the first computer entity should not be trusted.

31. (Currently Amended) A method of establishing trust between two computer entities, the method comprising:

[[the]] a first computer entity seeking a granting of trust from [[the]] a server by sending an inquiry in the form of a can-attest message to the server, the can-attest message stating that the first computer entity can send an attestation message but that the first computer entity would like to know from the server whether such an attestation message is required, and if so any requirements that such server has with regard to such attestation message;

the server sending an attestation-wanted message to the first computer entity in response to the can-attest message, the attestation-wanted message stating that the server does in fact require an attestation message from the first computer entity and that the attestation message as sent by the first computer entity must adhere to certain requirements as defined in such attestation-wanted message;

transmitting an attestation message from a first computer entity to a second computer entity, the attestation message including a code identifier (code ID) associated with the first computer entity that is calculated by using a security ID associated with the first computer entity and corresponding to a behavior parameter that is associated with a computing operation having security implications;

ensuring that the security ID corresponding to the behavior parameter has not been tampered with, by verifying the validity of the code ID in the second computer entity, the verifying comprising determining that the first computer entity is not included in a do-not-trust list;

transmitting a trust message from the second computer entity to the first computer entity upon successfully verifying the validity of the code ID, the trust message including a first secret

that is shared between the first and the second computer entities for communicating securely over a first period of time, wherein the first period of time is determined by the second computer entity, and

the security ID including security information relating to the first computer entity, the security information being expressed as a number of name-value security attribute parameters, the first computer entity being an executable and referring to the parameters in the security information in the security ID to determine whether particular security behavior is allowed, the code identifier (code ID) being representative of the first computer entity and calculated as a one-way hash of a combination of the executable of the first computer entity and the security ID so that modification of the security information in the security ID causes the calculated code ID to change and the second computer entity can interpret the change as an indication that the first computer entity should not be trusted.

32. (Previously Presented) The method of claim 31, wherein the security ID is stored in a location in the first computer entity, and wherein the first computer entity is constrained to executing a particular behavior only via accessing the stored location.

33. (Previously Presented) The method of claim 31, wherein the behavior parameter comprises at least one of a) opening of a file in the first computer entity or b) opening a debugging port in the first computer entity.

34-35. (Canceled)

36. (Previously Presented) The method of claim 31, further comprising:

retransmitting the trust message from the second computer entity to the first computer entity, the retransmitted trust message including a) a second secret that is different than the first secret, and b) data to inform the first computer entity of a second period of time over which the second secret is valid.

37. (Previously Presented) The method of claim 30, wherein the can-attest message is transmitted from the first computer entity to the server without encryption.

38. (Previously Presented) The method of claim 30, wherein each of the can-attest message and the attestation-wanted message is transmitted without encryption.